DOCUMENT RESUME

ED 121 786 SP 010 051

AUTHOR Norman, Donald A.

TITLE Studies of Learning and Self-Contained Educational Systems, 1973-1976. Report No. 7601.

INSTITUTION California Univ., La Jolla. Center for Human Information Processing.

SPONS AGENCY Advanced Research Projects Agency (DOD), Washington, D.C.; Office of Naval Research, Washington, D.C.
REPORT NO ARPA Order-2284
PUB DATE Mar 76
NOTE 53p.

EDRS PRICE MF-$0.83 HC-$3.50 Plus Postage
DESCRIPTORS *Computer Assisted Instruction; Concept Formation; Educational Programs; Learning; *Learning Processes; *Models; Systems Approach; *Teacher Education; Teacher Programs; Teaching Techniques; *Tutorial Programs

ABSTRACT

Studies of the teaching and learning process provide the basis for this research. The emphasis is on the nature of the tutorial interaction during the learning of complex material, with the eventual goal of being able to construct an automated, computer-controlled tutorial system. An active structural network (a type of semantic network) was used to represent both knowledge and procedures within the same data base. An event schema representation was developed to study the acquisition and organization of complex events. The schema was used to study students learning American Civil War history or how to program in the computer language FLOW. The studies of the learning and teaching processes are primarily concerned with developing models of the student and building automated tutorial instructional systems. The automated systems provide a controlled environment in which to observe learning and teaching, and the two topic areas of history and programming were used to ensure the generality of the models. The technique of incremental simulation, in which the computer performs whatever aspects of the tutorial project are fully understood and human tutors enact the roles of other system components, provides for the extension of understanding of the interactions among the components of the overall tutorial system. (Author/JMF)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED121786

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

STUDIES OF LEARNING AND SELF-CONTAINED EDUCATIONAL SYSTEMS, 1973-1976

CENTER FOR HUMAN INFORMATION PROCESSING
University of California, San Diego
La Jolla, California 92093

Sponsored by
Advanced Research Projects Agency
and the Office of Naval Research
ARPA Order No. 2284

March 1976

Report No. 7601

Approved for public release; distribution unlimited. Reproduction in whole or in part is permitted for any purpose of the United States Government.

SP 010 051

Studies of Learning and Self-Contained

Educational Systems, 1973-1976

Contract Number: N00014-75-C-0757

ARPA Order Number: 2284

Task Authority Identification Number: NR 154-360

Program Code Number: 5W10

Effective Date of Contract: 1 February 1973

Contract Expiration Date: 31 January 1976

Amount of Contract: $327,521

Organization: University of California, San Diego

Principal Investigator: Donald A. Norman

Phone: (714) 452-2947

Scientific Officer: Dr. Marshall Farr, Director
Personnel and Training Research Programs
Office of Naval Research (Code 458)
Department of the Navy
Arlington, Virginia 22217

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, the Office of Naval Research, or the United States Government.

This research was supported by the Advanced Research Projects Agency and the Office of Naval Research of the Department of Defense and was monitored by ONR under Contract No. N00014-75-C-0757.

Approved for public release; distribution unlimited.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 7601	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Studies of Learning and Self-Contained Educational Systems, 1973-1976·		5. TYPE OF REPORT & PERIOD COVERED Final Report 2/1/73 - 1/31/76 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Donald A. Norman		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0757
9. PERFORMING ORGANIZATION NAME AND ADDRESS Dr. Donald A. Norman Center for Human Information Processing University of California, San Diego La Jolla, California 92093		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 154-360
11. CONTROLLING OFFICE NAME AND ADDRESS Personnel and Training Research Programs Office of Naval Research (Code 458) Arlington, Virginia 22217		12. REPORT DATE March 1976 13. NUMBER OF PAGES 48
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Learning, teaching, memory, tutorial instruction.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This research has centered upon the study of the representation of knowledge in memory and of the learning and teaching processes for complex topic matters. An active structural network, a type of semantic network, has been used to represent both knowledge and procedures within the same data base. An event schema representation has been developed to study the acquisition and organization of complex events. In particular, the schema representation has been used to study students learning American Civil War

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

history or how to program in the computer language FLOW.

The studies of the learning and teaching processes have been primarily concerned with developing models of the student and with building automated tutorial instructional systems. The automated systems have given us a controlled environment in which to observe learning and teaching, and the two contrasting topic areas of history and programming have been used to help insure the generality of our models. By the technique of incremental simulation in which the computer performs whatever aspects of the tutorial project are fully understood and human tutors enact the roles of other system components, we have been extending our understanding of the interactions among the components of the overall tutorial system. The goal is to develop a theoretical model of the learning of complex material.

STUDIES OF LEARNING AND SELF-CONTAINED
EDUCATIONAL SYSTEMS, 1973-1976

Work on this contract has been concerned with studies of the teaching and learning process. The emphasis has been on the nature of the tutorial interaction during the learning of complex material, with the eventual goal of being able to construct an automated, computer-controlled tutorial system.

In order to understand tutorial instruction, one must understand how a person acquires new knowledge. Accordingly, we must come to understand the ways by which knowledge is represented within human memory, how new information is incorporated within the old knowledge structures, and how a person comes to use the new information in appropriate ways. The tutor must have knowledge of both the subject matter and the changing state of the student. If we intend to automate the instructional process, we need to be able to understand how a good tutor operates, and this requires us to understand and be able to state:

1. The representation of the subject matter knowledge within the human memory system.
2. A model of the student, which includes representing the knowledge and the learning process.
3. The principles and strategies of instruction.

How does someone learn a complex topic matter? Consider, for instance:

1) Learning the history of the American Civil War

2) Learning how to program a computer

3) Learning the psychology of hearing

These topics differ in style and form, but all have the common properties of any complex topic matter--each topic requires the integration of a vast amount of material. The most difficult part of the learning of each of these topics is in understanding the interrelationships of the components that make up each topic. The factual learning, or committing to memory of the individual components, is not any particular problem. The level of performance of students on tests of these topics is not much affected by allowing them free access to reference books during testing. The complexity clearly resides in the interrelationships rather than in the acquisition of the individual components.

The study of complex learning poses two types of problems. One is concerned with the psychological basis of knowledge, theoretical understanding of the learning process and of the knowledge structure within human memory. The other is concerned with the actual presentation of material to a learner and with assessing the learner's evolving understanding of the material.

Learning and Teaching Complex Material

One feature of the learning and teaching of a complex body of material is that the interrelationships are so rich, there is no single sequence of presentation that is adequate. To understand a given concept fully, one must understand its relationships to all other relevant concepts. But this means understanding those other concepts as well, and they in turn require

a knowledge of the concepts with which they interrelate. Therefore, complex material must be presented in a way that allows the student to develop some framework for relating the pieces of information to each other, and then covers the material again, perhaps in more depth, enlarging and enriching the framework of knowledge already acquired. The effective teacher or textbook author follows this approach, not necessarily being aware of the rules being followed. We have called this approach web teaching (Norman, 1973).

An observation of the process of learning complex topics suggests some important properties that any theoretical description must be capable of handling. The learning of complex topics has continuity, with the new knowledge being added to previously acquired knowledge. The knowledge can exist at many levels of specification. It is possible to grasp a topic briefly and quickly, or to elaborate upon it indefinitely. This implies that knowledge about a topic is incremental. New concepts can be added to the previous knowledge base almost indefinitely. Knowledge of complex topics is robust. This does not imply that things are never forgotten or that mistakes are not made, but rather that a basic structure for the material can survive even large errors or large gaps of knowledge. The robustness of complex knowledge seems to have two causes: 1) any given unit of knowledge is determined by multiple sources; 2) it is possible to derive many of the units from other things that are known. Thus, complex topics are indeed complex. They consist of an interlocking network of concepts, each of which contributes to the overall knowledge, but none of which is indispensible.

Studies of the Representation of Knowledge

To understand what happens when a person acquires knowledge of a complex topic, one must know of the operations of the human memory system, how material is entered into that sastem, how it is represented, and how it is later retrieved and used. These are complicated issues which are at the current state of the art in psychology. New developments in our understanding of memory structures have occurred in recent years, and it appears that we are discovering the basic structures of human memory, of the knowledge within it, and therefore, of the process of learning. Considerable progress has been made on this problem within the contract research period.

Semantic networks. A number of workers in the field of human memory have recently been developing formal representations for the knowledge within memory. Most of the models that we are interested in here are described as semantic networks and, following the work of Quillian (1968), are all characterized by a directed, labelled graph structure. Carbonell (1970) used semantic networks for the knowledge base in a computer-assisted instructional system, and the work has continued under the direction of Collins. The version we have helped develop is called an active structural network to emphasize that the representation is both active and passive, and that it can contain general procedures that can be executed whenever functional knowledge must be used. (This work is reported in Norman, Rumelhart and the LNR Research Group, 1975; we will refer to it as the LNR work.) Although we base our discussion on the LNR work, in fact the comments apply to all semantic network representations.

Basically, the semantic network provides a means of representing knowledge. It is a new tool in psychology. Previously, our formal models

have been abstract. For example, a mathematical learning model predicts the probability of the strength of some association between two elements, but the elements are usually part of a large homogeneous set. With semantic networks, we look at the structure of very particular items.

The internal components of verbs. Consider how a child might come to learn language. In the work of the LNR Research Group, verbs can be decomposed into underlying primitive elements. Thus, some verbs specify only STATIVE components; others are more complex, specifying CAUSE and CHANGE. The statement that

(1) The skier went to the top of the mountain.

is represented as shown in Figure 1, which shows the decomposition of the verb "went" into its more basic underlying components.

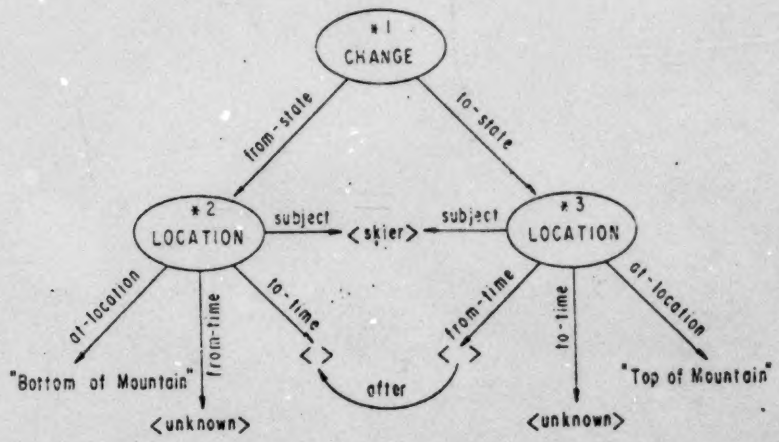


Figure 1. The skier went to the top of the mountain.

Network structures are composed of nodes and ordered, labelled relations connecting these nodes. In Figure 1, the ovals represent nodes that are token instances of propositional structures, the angular brackets (e.g., <skier>) represent nodes that are token instances of concepts, and the phrases in quotation marks (e.g., "bottom of mountain") represent network structures for the concepts described by the phrases, but which we have not shown in detail in the figure in order to preserve the diagram's clarity. (The numbers of the token propositional nodes have no intrinsic meaning: they are used to facilitate reference to them. Node *1 is called "node star 1," for example.)

Figure 1 can be interpreted in a straightforward manner by starting with node *1, the oval labelled CHANGE. This indicates that some change of state has taken place: the relations leaving node *1 describe the states that are involved. The CHANGE takes place from a state shown by node *2, which says that its subject (a skier) was located at the "bottom of mountain" from some unknown time to some time not specified (but indicated by the unnamed node shown as angular brackets). The result of the change is the state represented by node *3: the skier is now at location "top of mountain." Notice that the time he was located at the top of the mountain is not specified, except to indicate that it is later than the time at which he was no longer at the bottom of the mountain.

A more complex verb is one that implies causality: for example, "give." The statement

(2) Suzette took the skis from Henry.

means that Suzette did something that caused the skis to change from the possession of Henry to herself. The structure for sentence 2 is shown in Figure 2.

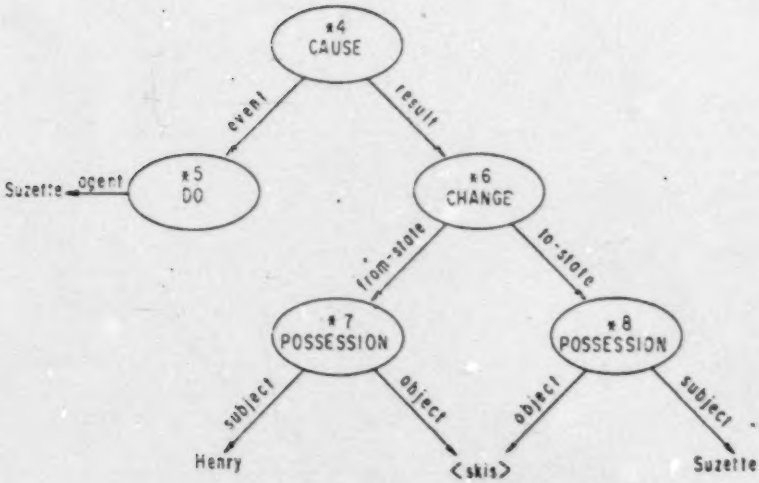


Figure 2. Suzette took the skis from Henry.

Note that these structures provide frameworks for adding later knowledge. Each structure is a memory schema: it allows us to organize material that is learned later on. The statement

(3) She promised to return them by morning.

adds information to our previous picture. Now we know that Suzette has an obligation to return the skis. We can deduce that she got the skis from Henry with his permission, and so we can expand the framework for the knowledge to something like that shown in Figure 3.

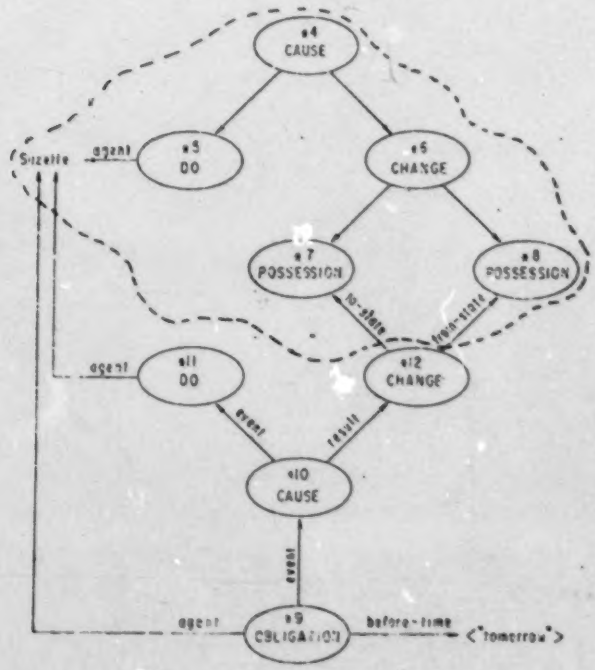


Figure 3. Within dotted line: Suzette took the skis from Henry. Outside dotted line: She promised to return them by tomorrow. Total structure: Suzette borrowed the skis from Henry promising to return them the next day.

Note that we add the new knowledge directly to the framework for the old. Thus, not only did we have a convenient way of modifying the structure for the previous episode according to the framework provided by the schema for the first sentence, but we have now modified the structure into one that is equivalent to the concept of "borrow": Suzette borrowed the skis. It is easy to see how similar statements of added information could have modified the structure to indicate that:

(4) Suzette stole the skis (no permission was granted).

(5) Suzette purchased or rented the skis (she paid money for them).

(6) Suzette got the skis by asking Peter to pick them up for her (expanding the DO statement).

The representation of simple sentences such as these is only a very small step toward the representation of complex topic matters. But it provides a start.

To answer a question intelligently requires a large body of specific knowledge about the area being questioned, as well as general knowledge about the world and its causal and physical laws, and also an understanding of the knowledge and behavior of other people. Those of us trying to model the human use of memory have just barely grown used to the fact that we must add knowledge of the world to the model, so it comes as a surprising and new challenge that we must also incorporate a person's understanding of other people into the model.

A parallel development is occurring in linguistics. The linguist is realizing that real world knowledge and an interpretive ability which ties

general social and linguistic knowledge to specific situations must be included in speaker-hearer models to account for the language user's ability to produce and understand sentences (Mehan (1972). A model which only incorporates formal grammatical considerations simply cannot account for the complexities of language usage in interactional situations.

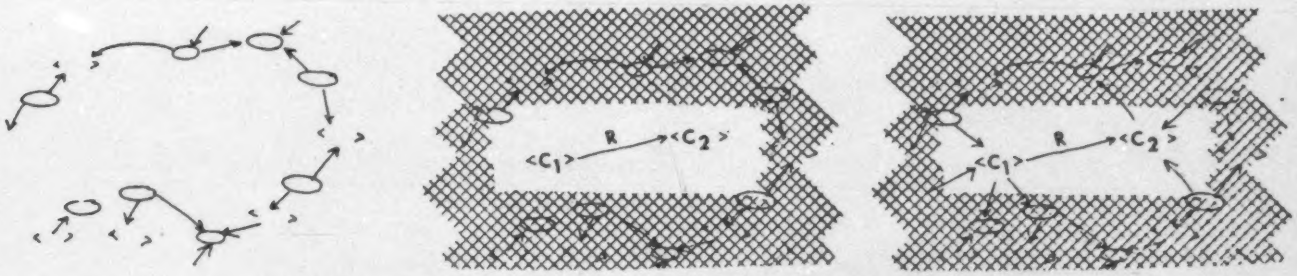
We have examined a number of different features about the representation of information in memory. An active representation seems to be necessary, with previously acquired knowledge about the world providing a structural framework upon which to construct new knowledge. The pre-existing structure not only defines the way that new information will be represented, but it helps in organizing the information that is not yet known. Thus, the structure can maintain open positions within its framework for the necessary causes and results, objects and actors, and even methods and actions that may remain unspecified. In this way, students acquiring knowledge can be led to seek more information to fill the missing nodes in their developing memory representations.

Learning

Teaching and learning are the names that we give to the activities of presenting and acquiring new information. There is no adequate theory of instruction to guide us in these activities, but an examination of the structure of memory from the viewpoint represented here appears to provide useful guidelines for the development of such a theory. Indeed, MacDonald-Ross (1972) has studied network relationships with just this goal in mind, to determine how to design an educational system for the Open University in England. MacDonald-Ross argues that structures of this nature do guide both students and faculty into an understanding of their subject matter.

Examine the structure of memory. It is an interconnected network, with new facts supported within the skeletal structure provided by the old. To acquire new information means to construct new nodes and the relations between them. Consider how that might be accomplished. First, let Figure 4A represent a segment of knowledge in the memory. If we acquire two new concepts (C_1, C_2) and a relation between them (R) as shown in Figure 4B, there are no connections to the old network. Retrieval should be difficult, and perhaps even the acquisition is difficult. This is poorly learned information.

Now consider Figure 4C. Here, the two newly acquired nodes are well integrated. To use an obvious analogy drawn from the structure itself, we would say that the newly learned components of 4C are well supported; those of 4B lack support. Notice, too, that 4C is better integrated within the network, in part because it contains more information. If this view of things is correct, it explains why mnemonic techniques are so useful in causing arbitrary strings of items to be learned even though they add to the total amount of material that is to be acquired. They provide a firm, well integrated structural framework.



(A) (B) (C)

Figure 4. A network representation of the knowledge structure in memory.

Teaching

Suppose we have a large body of knowledge to teach--what would be the best way of doing it? Presumably, we need to interconnect the new information with the existing structure. One way to do this is to construct a supporting web structure first, and then fill in the details. To do the details first would not work, for without a supporting structure, the new material simply could not become integrated. In teaching, this means that an outline of the material to be learned should be acquired first, then a more detailed overview, and then progressively more and more detailed structures.

The network representation of knowledge can guide the process of instruction in two different ways: 1) if we have a good representation of the knowledge we wish to teach, then we can organize it properly for efficient learning; 2) if we try to discover the network representation of the student, we can use this to guide our teaching. Knowing the knowledge structure of the student helps in devising the original level of organization of the material. In addition, as the lessons progress, we can use our understanding of the student's developing structures to guide us in teaching, telling us what old material has not been acquired and what new material might perhaps already be known. Thus, it is theoretically possible to tailor instruction to the knowledge base and competence of the student.

Whether the network representation makes the goal easier to attain remains to be seen. The major drawbacks have resulted from the expense of using tutorial methods in mass education, and from the lack of sophistication in the implementation of most teaching machine programs and computer-assisted instructional systems to attempt these goals (with the exception of the

demonstration systems of Brown, Burton & Zdybel, 1972; Carbonell, 1970). Nonetheless, the analysis is instructive even if the complete implementation remains in the future.

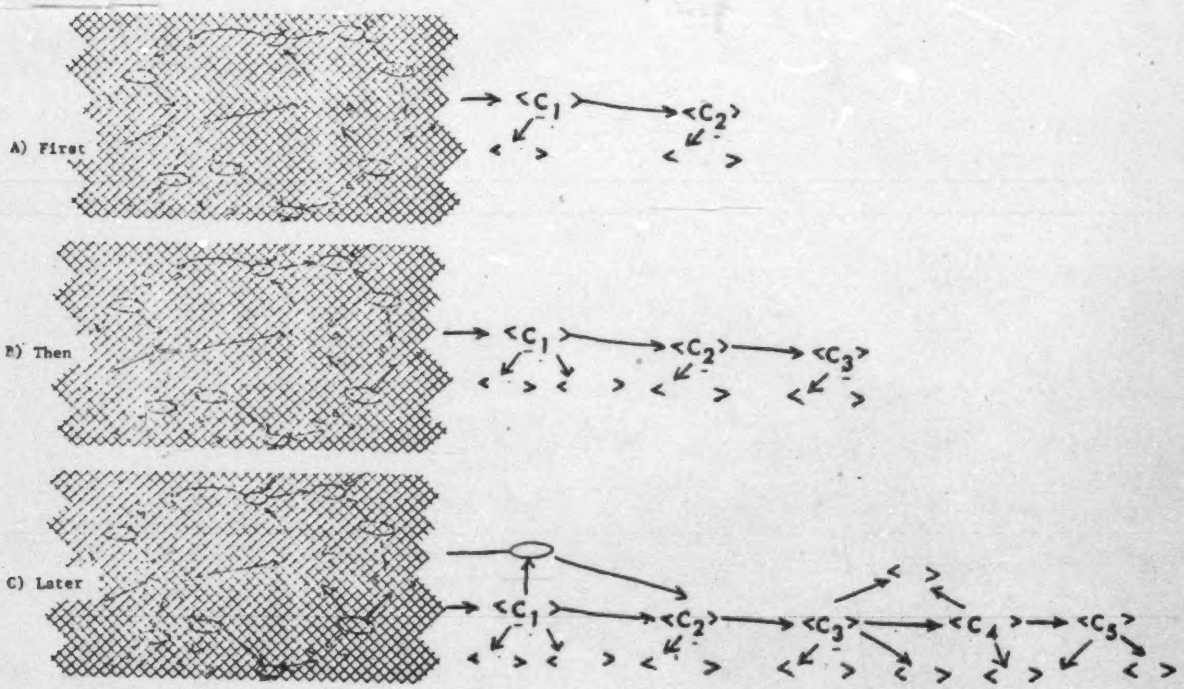
We can characterize two different strategies of presenting material: two different strategies of teaching. One is to present a cohesive organized structure to the student, carefully adding one new piece of information after another to the developing structure. This might be called linear teaching. It is the system the characterizes lectures, textbooks, and even the structure of this report. The other method is to present a coarse web of information, outlining the topics to be discussed, then giving a general overview followed by more detailed overviews, and finally the detailed substructure. This procedure might be called web teaching. Web teaching is often prescribed, seldom done. It is difficult to perform well. But we wish to suggest that for the learning of complex topic matters, web teaching may at times be more efficient.

The distinction here between web and linear strategies are closely related to, yet different from, the distinction between holistic and serialistic strategies of learning (Pask, 1971; Pask & Scott, 1972). Pask and Scott gave students freedom to explore a network of knowledge about probability theory. Some students, the serialists, preferred to work on one node at a time, always completing that node before proceeding. Others, the holists, worked on many nodes simultaneously. These are interesting results that, hopefully, can be used to characterize styles of learning.

Linear teaching. Linear teaching (and its complement, linear learning) might be characterized by the sequence shown in Figure 5. At first there is

a well developed structure structure of knowledge to which we add a new set of nodes, say the linear string C_1 and C_2 and their subnodes, as shown in Figure 5A. Further learning adds another node and more subnodes (Figure 5B). Eventually a longer linear string is acquired, nodes C_1 through C_5 in Figure 5C. Presumably, as learning increases the older material becomes more thoroughly embedded into the original knowledge structure, but the overall picture remains one of a systematic linear increase in the knowledge base. The new knowledge is weakly supported. If one link in the chain is lost, either because it was poorly acquired or because the learner might have missed exposure to that part of the material, then the rest of the structure is weakened and may easily become irretrievable. This is a familiar experience to many instructors—a student who failed to learn an important point made early in a course may thereafter be in difficulty, unable to understand most of what has followed from it.

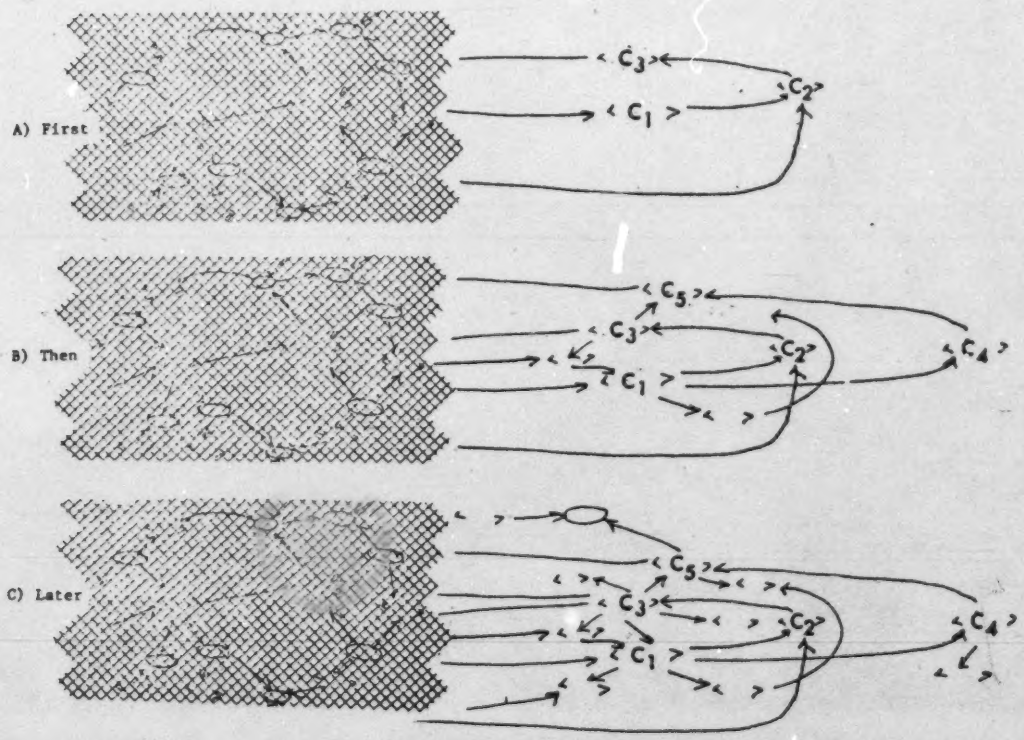
Figure 5. Linear learning.



Web teaching. Web teaching (and web learning) might be characterized by the sequence shown in Figure 6. Here, we use the first few nodes acquired (Figure 6A) to establish a coarse web which is well integrated with the previous knowledge structure. Then we insert refinements within the structure created by the original web framework, as shown in Figure 6B. Finally, we can fill in the details, as shown in Figure 6C. In web learning, no single node is critical to the whole, so that poor acquisition or even the absence of a single set of concepts does not destroy the validity of the structure.

The structures described in these two diagrams (Figures 5 and 6) are at best weak analogies to the processes of learning and teaching. As yet, they are far from being a formal, testable theory. But the analogies have led to useful insights at this early stage of development in the planning of course material. Whether there is more to the distinctions between web and linear teaching than superficial analogies remains to be investigated.

Figure 6. Web learning.



In examining how a person learns a complex topic matter, we face a major difficulty with the active structural network (and with all work based on a semantic network). The representation as originally formulated did not readily allow for different types of knowledge. That is, with the representational system it is possible to represent individual events and concepts, but not easy to represent higher level conceptualizations, or causes of those events. For example, it is easy to represent that General Grant smoked cigars or that the Amazon River had numerous tributaries, but not easy to represent that the North's plan of action in the Civil War was to cut off the South from external supplies, and that this plan motivated Lincoln to keep a major segment of troops at Cairo, Illinois and St. Louis, and that it eventually led Grant to cut off the Tennessee and Cumberland rivers. High level summaries of plans and goals that motivate behavior must be part of any knowledge base about the war. At the time the contract research was started, little was known about the representation of causes, motives, plans, and goals, making it impossible to represent complex interrelations among the events that characterize most real situations.

Event Schemas

In 1974, Rumelhart developed a representational scheme for handling causal events within the framework of the LNR structural network (described in Rumelhart, 1975). Rumelhart's work was important in revealing the underlying structure of complex episodes, and we were able to adapt the story schema that he had originally intended for use in the analysis of

fairy tales to more complex material--in particular, to event schemas appropriate for representing the American Civil War.

Many stories have a readily apparent structure: first an original setting for the story is presented, and then an episode. Within the episode, there is a series of events that befall the protagonist, and each event initiates some reaction. These aspects of stories can be captured by simple functions (which we call predicates) that relate settings to episodes, events, and goals. Hence, a story can be decomposed into a setting that "allows" an episode:

(7) ALLOW (setting, episode)

Similarly, an episode occurs when some event initiates a reaction:

(8) INITIATE (event, reaction)

In addition, we have causes and motivations to consider, complete with internal responses of the actors and their overt responses as they react.

An actual historical event, such as the American Civil War, is more complex than stories in that there are numerous protagonists, each often with independent and conflicting motivations. The interactions of these protagonists are the causal events that trigger a good deal of the war action. Sometimes the events are premeditated, sometimes they result from the accidental congruence of the actors. The basic ideas of the story schema are compatible with events such as the Civil War even if the details are not. With the assistance of Rumelhart, we developed an event schema representation of the Civil War.

The Civil War representation structures the aims, goals, and motivations of the protagonists of the war. It enables us to represent such issues as:

(9) Why was Fort Henry important?

(10) What was the southern strategy in 1862?

(11) Where was the southern line of defense after the Battle of Shiloh?

In addition, because of its heavy emphasis on causal factors, the Civil War structure enables the answering of causal questions at several levels of analysis:

(12) Question: Why did Grant capture Fort Donelson?

Answer 1: Because he bombarded the Fort from both land and sea.

Answer 2: Because he wished to drive the Confederates from the Fort.

Answer 3: Because he was only 12 miles away, at Fort Henry.

Answer 4: Because Fort Donelson was on the Cumberland River.

Answer 5: Because the Cumberland River was an important transportation route.

Answer 6: Because it was important that the North gain control of the western rivers.

Each of these different answers is correct, but which is appropriate depends upon the particular situation. With the event schema representation, all of these answers are in the data base, and the only problem is to develop the strategies for determining which levels of analysis are appropriate at any given time.

The Civil War Tutorial System

Consider how such a knowledge representation might be used. Figure 7A shows a small section of the data base that represents the Civil War battles around the forts on the Tennessee and Cumberland Rivers (Forts Henry and Donelson). The data base was encoded according to the event schemas appropriate to the Civil War. To create the data base, we used the MEMOD computer system which we have developed (Norman, Rumelhart, & the LNR Group, 1975), using the SOL language parts of that system to define verbs and other words that create the appropriate structures. Then we constructed the text shown in Figure 7B and typed it into the system, which automatically created the appropriate memory structures.

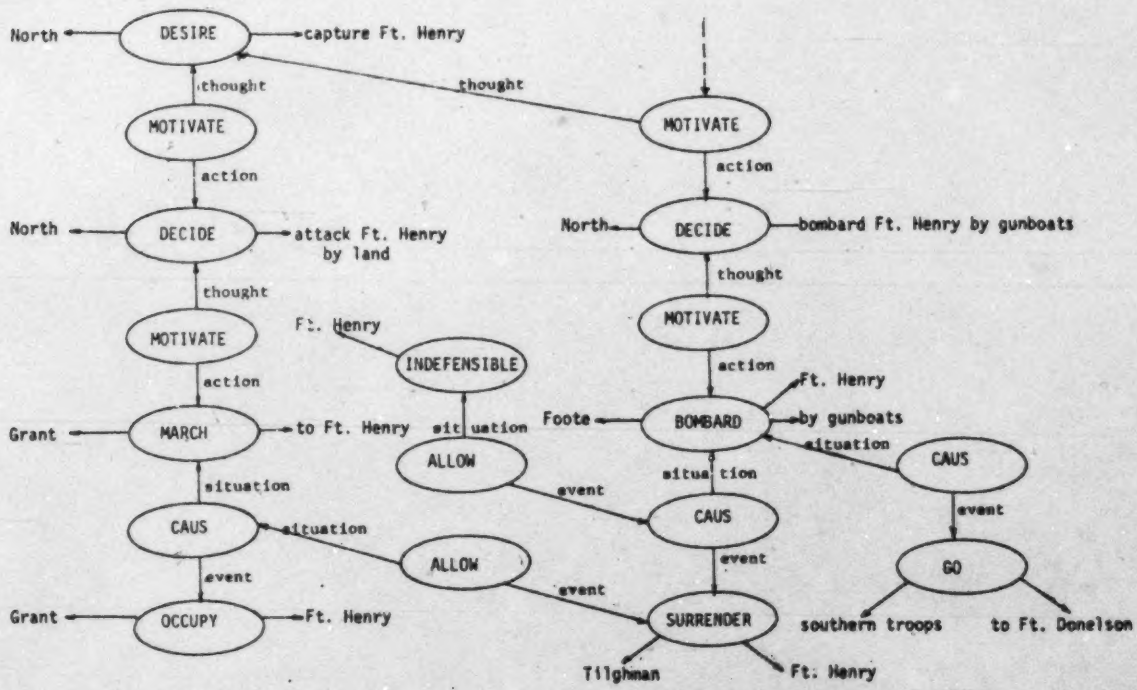


Figure 7. The data base.

A) Schematic representation of the data base.

```
THE DESIRE THAT THE-NORTH CAPTURE FORT-HENRY MOTIVATES THAT
THE-NORTH DECIDE TO ATTACK FORT-HENRY BY LAND.
%
     CONNECT THE "DECIDE" WITH "TEXT" TO
     "THE-NORTH DECIDES TO ATTACK FORT HENRY BY LAND".

%
%
THE "DECISION" MOTIVATES GRANT TO MARCH TO FORT-HENRY.
%
     CONNECT THE "MARCH" WITH "TEXT" TO
     "GRANT MARCHES TO FORT HENRY".

%
%
THE "MARCH" CAUSES GRANT TO OCCUPY FORT-HENRY.
%
     CONNECT THE "OCCUPY" WITH "TEXT" TO
     "GRANT OCCUPIES FORT HENRY".

%
%
THE DESIRE THAT THE-NORTH CAPTURE FORT-HENRY MOTIVATES
THE-NORTH TO DECIDE TO BOMBARD FORT-HENRY BY GUNBOATS.
%

     CONNECT THE "DECIDE" WITH "TEXT" TO
     "THE-NORTH DECIDES TO BOMBARD FORT HENRY BY GUNBOATS.

%
%
THE "DECISION" MOTIVATES FOOTE TO BOMBARD FORT-HENRY BY GUNBOATS.
%
THE BOMBARDMENT CAUSES TILGHMAN TO SURRENDER FORT-HENRY.
%

     CONNECT THE "SURRENDER" WITH "TEXT" TO
     "TILGHMAN SURRENDERS FORT HENRY".

%
%
THE FACT THAT FORT-HENRY IS INDEFENSIBLE ALLOWS THE "CAUS".
%
     CONNECT THE "INDEFENSIBLE" WITH "TEXT" TO
     "FORT HENRY IS INDEFENSIBLE".

%
%
THE "SURRENDER" ALLOWS THE "MARCH" TO CAUS GRANT TO OCCUPY FORT-HENRY.
%
%
THE BOMBARDMENT CAUSES SOUTHERN-TROOPS TO GO TO FORT-DONELSON.
%
     CONNECT THE "GO" WITH "TEXT" TO
     "THE SOUTHERN TROOPS GO TO FORT DONELSON".
```

B) Text used to create the data base.

The tutorial interaction was performed with the aid of the computer system through a technique called "incremental simulation." We automate as much of the system as is possible, and allow humans to enact the roles of those components of the system not yet automated. This technique enables us to test our ideas before all the details of the system are fully understood. It also permits us to make changes more rapidly as we learn about the component interactions than would be possible with full simulation. Figure 8 (A and B) presents the conceptual models of the student and tutor. In addition to the data base, there is a set of routines to help the tutor interact with both the data base and the student. The instructor and the student sit at computer terminals and communicate through the computer. The student can talk only to the instructor, but the tutor can talk to the student or to the different parts of the system. These communication pathways are illustrated in Figure 8C.

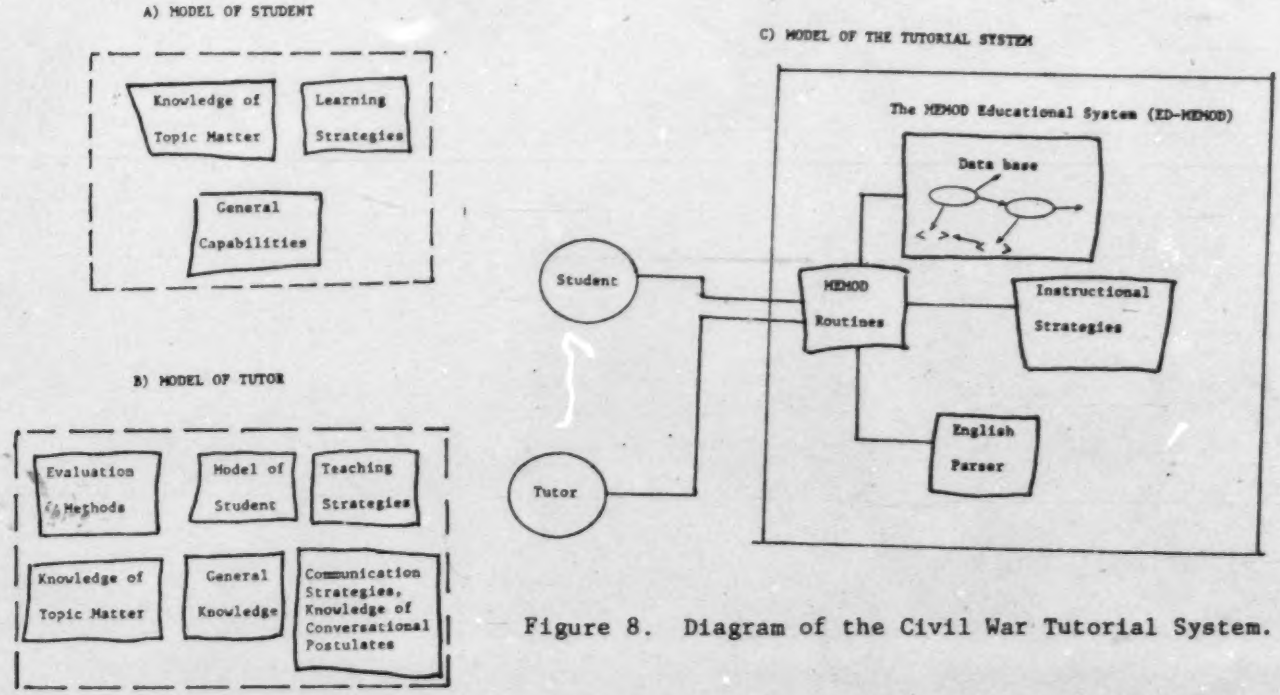


Figure 8. Diagram of the Civil War Tutorial System.

The tutor can ask questions, evaluate the student's answers, and present information in several ways. The tutor can discuss a topic by finding one piece of information in the data base and travelling along the links of the network to get to other pieces. This can be done one link at a time or by using a semi-automated version that will continue along a causal chain asking questions. (These network-travelling routines are an initial step toward the development of automated teaching strategies.) The tutor may also supply a piece of information directly by typing it in rather than using the data base.

Currently the system leaves control and initiative up to the tutor. It does not prompt to ask or present or evaluate, although an evaluation will be required by the routines that encode the model of the student. A sample (simulated) dialogue using the system is shown in Table 1.

Table 1. Simulated Dialogue

TUTOR TO SYSTEM	SYSTEM TO STUDENT	STUDENT TO TUTOR
1. STATE "*DESIRE-3016".		
	2. THE NORTH DESIRED TO STOP TRAFFIC ON THE WESTERN RIVERS.	
3. FOLLOW "MOTIVATE".		
	4. WHAT DID THE NORTH DECIDE TO DO THEN?	
		5. ATTACK FORT HENRY.
6. RIGHT.		
	7. RIGHT.	
8. THE NORTH DECIDED TO ATTACK FORT HENRY BY BOAT AND BY LAND.		
	9. THE NORTH DECIDED TO ATTACK FORT HENRY BY BOAT AND BY LAND.	
	10. WHO MARCHED TO FORT HENRY?	
		11. FOOTE.
12. WRONG.		
	13. WRONG.	
14. GRANT MARCHED TO FORT HENRY.		
	15. GRANT MARCHED TO FORT HENRY.	
16. STATE "*SURRENDER-3134".		
	17. TILGHMAN SURRENDERED FORT HENRY.	
18. GO UP A LEVEL, THEN GO UP ANOTHER LEVEL AND ASK.		
	19. WHAT DID FOOTE DO?	
		20. SHELLED FORT HENRY FROM GUN BOATS.
21. RIGHT.		
	22. RIGHT.	
		23. ASK TUTOR WHERE IS FORT HENRY.
24. STATE "*LOCATE-2989".		
	25. FORT HENRY IS ON THE TENNESSEE RIVER. FORT DONELSON IS ON THE CUMBERLAND RIVER.	
26. STATE "LOCATE-3217".		
	27. FORT HENRY IS ON THE TENNESSEE, ABOUT 12 MILES FROM FORT DONELSON, AND ABOUT 100 MILES FROM CAIRO, ILLINOIS.	

Summary of the Civil War Tutorial Program

We have been able to represent a considerable segment of the events and conditions of the Civil War in the computer data base, including the causal structure of the actions during the war. We are able to use the data base to interact with a student by using a mode in which a human tutor directs the system how to progress through the data base. This tutorial system has met a number of our basic objectives, although it is not automated and still requires a human tutor.

The basic premises underlying the tutorial strategy just described are:

1) The tutor assesses the knowledge of the student and constructs a model of the student's knowledge of the topic matter.
2) The tutor compares the knowledge of the student with the topic, marking the gaps and inconsistencies.
3) The tutor presents the material needed to fill in the gaps in the student's knowledge.
4) The presentation of new material follows some basic organizational principles in terms of web teaching strategy.

Studies of Tutorial Interaction in FLOW

We have examined in detail the developing representational structure of students learning FLOW, a simple algebraic programming language. This work has been reported in technical reports (Norman, 1973; Norman, Gentner & Stevens, 1974) and as a book chapter (Norman, Gentner & Stevens, 1976). The most important issues concern the representation of the knowledge of the student. We believe it best characterized by means of memory schemas, small, self-contained units of procedural knowledge. We showed that when students encounter a new concept, such as the JUMP-TO instruction in FLOW, they impart some structure to their interpretation of that instruction. When the structure proves to be wrong, then the students must use their knowledge of the behavior of the program and the discrepancies from their predictions to modify their schemas. The process s iterative, with each cycle taking the students closer to the accurate representation of the instruction. In all of this, the schema is the unit that is of interest, representing the end product of the learning sessions.

A schema is a basic unit of knowledge. It provides a framework upon which different elements of information about a topic can be integrated. Presumably, schemas are small in size, probably limited by the amount of material that can be held within short-term memory. Thus, schemas must refer to other schemas in order to cover any reasonable amount of material. When schemas refer to one another, we have proposed that they do so through context-dependent descriptions (Bobrow & Norman, 1975). By this we mean that the relations among schemas are relative to the context in which they are used, so that different contexts may very well use different schemas to fill the same description. This allows for the use of matephorical structures.

The important lessons learned from our studies of schemas were that in teaching, one must build upon a framework of material. Students must understand the limitations of their own knowledge, and they must learn to evaluate and modify their own schemas.

FLOW is a simple, interactive computer language whose commands are designed for string manipulation. FLOW is specifically aimed at the beginning computer programming student, and has a number of advantages as a subject matter for use in studies of learning. Like many other common computer languages, FLOW involves both conceptual and procedural knowledge. It is problem oriented. There is a large variety of problems that can be posed to exercise the student's developing knowledge. The language itself and the tasks form a small, concise, and well defined body of knowledge. The statement of a problem and what consitutes an acceptable solution are normally fairly clear, although there may be some debate about the quality of a particular solution.

The students in most of our studies were university undergraduates, and it is relatively easy to find subjects from that group with little or no experience with computer languages. Since FLOW is quite a simple language, a lot of interesting learning takes place within the first hour or two. The subject matter is complex enough, however, to require up to 10 hours to master. This forms a convenient time span for experimental studies. Subjects usually enjoy learning FLOW with its interactive nature and low level of frustration, and are highly motivated to learn.

The FLOW System

The FLOW system is based around a minicomputer with a CRT display terminal at which the student works. It is connected to the Burroughs 6700 at the University of California, San Diego Computer Center. This large computer contains other programs, such as the automated tutor, which interact with the student. There are facilities for recording complete protocols of experimental sessions including displays on computer terminals as well as voices. If the experimenter/tutor wishes to work from another room, a television monitor duplicates the display on the student's terminal, and there are terminals for typed communication with the student.

The FLOW Tutor

One of our projects which utilized the FLOW system was the development of a computer-based, automated tutor which assists a student who is learning FLOW. (The FLOW tutor has been described in detail by Gentner, Wallen & Miller, 1974.) Our basic purpose was to investigate how a tutor follows a student's progress and helps when there is difficulty. The method we used was to construct a computer simulation of a human tutor who watches over the shoulder of a student learning FLOW from a printed instruction booklet.

In operation, the student worked at a computer terminal using an instruction booklet containing a series of programming problems along with descriptive text and practice examples. The FLOW tutor monitored the student's keyboard inputs and continually updated a simple model of the student progressing through the instructions. The student model consisted of a series of states corresponding to sections of the instruction booklet. When the student worked on a problem, the tutor used an example of a correct solution to the programming problem and generated advice to the student based

on a comparison of the student's program and the correct program. (If there were several correct programs it arbitrarily used a simple and straightforward version for the comparison.) Whenever the student asked for help or ran a program, the tutor matched the output of the student's program against the output of the correct program. If the results of the student's program were correct, the tutor congratulated the student and suggested starting the next unit. If the results were not correct, the tutor compared the statements in the two programs and advised the student of the discrepancies.

The Conceptual FLOW Tutor

We have begun development of a new version of the automated tutor, known as the Conceptual FLOW tutor. The instructional context shown in Figure 9 is similar to that of the earlier automated FLOW tutor: the student reads an instruction booklet and works at a computer terminal while the tutor monitors the student and gives advice if the student appears to be in trouble. The Conceptual FLOW tutor, however, differs from the earlier

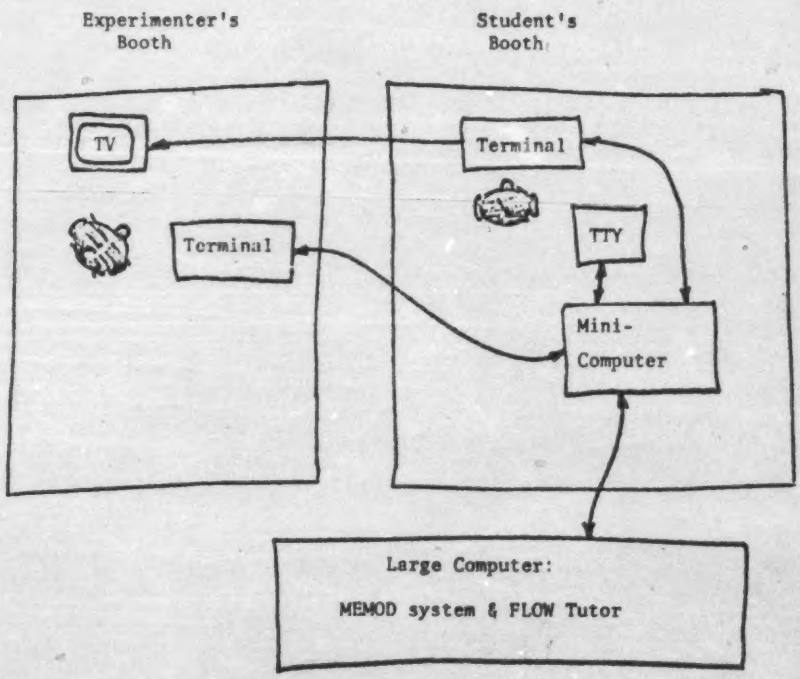


Figure 9. The FLOW Conceptual Tutor System.

tutor in a number of important ways. 1) The analysis of the student's work and the advice to the student can be handled in terms of higher level programming concepts such as loops and order of execution, while the original automated tutor worked only in terms of the FLOW statements. 2) The Conceptual FLOW tutor builds a more complex model of the student and uses that model to formulate advice to the student. 3) The tutor uses schemas to represent the knowledge data base, to solve problems, and to interpret student behavior.

In general, FLOW programs are understood in a hierarchy of four levels. At the highest level there is a description of the function or output of the program. The second level describes the topological structure of the program (this is roughly equivalent to a flow chart of the program). The third level consists of the actual statements in the program. The fourth and final level is the list of keys which must be pressed by the student to enter the program.

Typically, the tutor solves problems by starting with a description of the desired function and then determining the required FLOW statements and keypresses. The student's responses are interpreted by moving up in the hierarchy from keypresses or statements toward the overall function of the program. In operation, the tutor moves freely up and down the understanding hierarchy. For example, if the student has been given a programming problem which requires a sequence of actions to be repeated, the tutor realizes that this requires a loop, and that this requires a JUMP statement. The tutor examines the student's program for the appropriate JUMP statement. If the statement is missing, the tutor explains the need for a loop. Here the tutor has started at the top level, dropped down two levels to find a fault in the student's program, and finally moved up one level to describe the fault to the student.

Schemas and Their Uses

Schemas as used in the Conceptual FLOW tutor consist of a series of slots, each with a name and a value. Table 2 shows some typical schemas used by the tutor. The first schema in Table 2 is for the display statement. The FUNCTION slot of this schema has a value of DISPLAY, indicating that the function of the display statement is described by the higher-level schema for DISPLAY. The STATEMENT-NUM and STRING slots are now occupied by variables, but they would be filled in for any particular instance of a display statement. The METHOD slot on this schema has the value PRESS-3, an instance of the PRESS schema. By examining these schemas the tutor can tell that to get a display statement, a particular series of keys must be pressed.

The tutor must have considerable knowledge about programming in general and the FLOW language in particular. The tutor uses its knowledge to direct its operations. In the concept-driven mode, the tutor scans through its representation of the instruction booklet until it finds the next problem, deduces a correct solution for the problem, and predicts the next response of the student. If the student gives the expected response, the tutor notes that and goes on to predict the next response. An unexpected student response puts the tutor into data-driven mode, where it uses its knowledge to interpret the student's action. Some actions by the tutor may involve both modes of operations, as when the tutor interprets an unexpected student response as a typical error in the data-driven mode, then uses the high level schema for that error to predict the next student response in the concept-driven mode, and finally checks the student response to verify its hypothesis.

Table 2. Typical Schemas Used by the Conceptual FLOW Tutor

DISPLAY-STATEMENT ISA TYPEFRAME FUNCTION DISPLAY STATEMENT-NUM VARIABLE-2 STRING VARIABLE-1 METHOD PRESS-3	DISPLAY-STATEMENT is the name of the schema. DISPLAY is the name of another schema. VARIABLE-1 and VARIABLE-2 get filled in when the schema is used.
VARIABLE-2 TYPE VARIABLE CONSTRAINT 3DIGIT-NUMBER	This schema indicates that VARIABLE-2 must be a 3-digit number.
PRESS-3 TYPE PRESS KEY ORDER-1 FUNCTION DISPLAY-STATEMENT	This schema explains the METHOD of the DISPLAY-STATEMENT.
ORDER-1 TYPE ORDER ISAINVERSE D ISAINVERSE QUOTE ISAINVERSE VARIABLE-1 ISAINVERSE QUOTE KEY-I PRESS-3	This schema indicates that the display statement requires the student to press the following keys: D "...VARIABLE-1..." VARIABLE-1 is any string of characters.
VARIABLE-1 TYPE VARIABLE --- ---	

The Student's Acquisition and Use of Schemas

Just as a tutor has memory schemas that can be used to direct the flow of teaching, students too have schemas that guide their operations. Our studies of the processes of the students indicate that they start with conceptualizations that are then modified as they progress through the learning sequence. Students are active processors of the information presented to them, and the instructor must take this into account. Accordingly, we must learn to structure our learning strategies in relation to the interpretations and prior schemas of the students. A major difficulty in the teaching process is to get the students to realize their own misconceptions. Many times we have discovered that students have schemas for concepts, but that these schemas are not complete, or are otherwise erroneous.

In this section we illustrate some of the stages in the developing knowledge schemas of a student who was learning FLOW. In this example, we were not using the automated tutorial program, but rather were using a technique developed to give better data on the status of the student's knowledge. The student was instructed to attempt to predict the behavior of a program just prior to executing it, and she was continually asked to describe her thoughts, particularly when her predictions failed to be verified. We pick up the experiment from the point where the student is learning to use the command "Jump to." She has just previously learned to use the command "Print" in several different programs, but no program was longer than two lines and most contained only a single use of the "Print" command.

We start observing the student at the point where she has been asked to type Program 1 onto the display terminal.

Program 1: 010 Print "Rochelle"

020 Jump to 010

Experimenter: This program will make the computer repeat the printing of the word "Rochelle." What do you think the output will look like?

Student: The computer will print the word "Rochelle" twice.

The answer is consistent with the ordinary sense of the word "repeat." It is also consistent with the student's prior experience, for in previous programs with no JUMP statements and at most two PRINT statements, any program that repeated the same printout printed the same word twice. If we could characterize this student's schema for the purpose of the "Jump to" instruction, it probably would look something like this:

Schema 1: If the instruction is "Jump to n", then the computer does instruction number n.

Now the student was instructed to run Program 1. When she did so, the output that appeared looked like this:

RochelleRochelleRochelleRochelleRochelleRochelleRochelleRochelleRochelleRochel

Student: I guess it keeps repeating until someone tells it to stop.

By her comment, the student has clearly learned something more about the JUMP statement. To test what she had learned, we asked her to enter Program 2 into the computer and to predict its outcome.

Program 2: 010 Print "Hi"

020 Print "Rochelle"

030 Jump to 010

Experimenter: What do you think this program will do?

Student: Its first instruction is to print "Hi" so it will do "Hi", then it will (pauses) there's no space, so it will just go "HiRochelle" for the second instruction. And then it will go back to the first instruction which was Print "Hi", so it will just write "Hi" until we tell it to stop.

We see from this example that the student has modified Schema 1 into a new form, something like this:

Schema 2: Do each instruction in order unless the instruction is a JUMP-TO. If the instruction is JUMP-TO n, then continue doing instruction n until told to stop.

Note that this schema, even though incorrect, is perfectly consistent with everything the student has seen up to this point. She has derived her notion of sequential order of execution from earlier programs and has used it here to predict the first two elements of the output. From Program 1 she has seen that the JUMP-TO in that program caused the instruction to be repeated. Hence, she developed Schema 2.

The test of the student's schema came when she was asked to run Program 2. Here is what happened:

HiRochelleHiRochelleHiRochelleHiRochelleHiRochelleHiRochelleHiRochelleHiRochel

Once again the result was not what was expected. Once again the schema for "Jump" had to be modified.

Student: When you say "Jump to the first instruction", it will go to that and then I guess it goes to the second one and if there isn't a second one it will just keep repeating the first one. Otherwise it will repeat both.

This is a rather complicated and highly conditionalized notion, but it is perfectly consistent with all example she has seen. When she was asked to describe how the computer actually performed Program 2, she provided a correct line-by-line description. Her schema now might be characterized as:

Schema 3: Do each instruction in order unless the instruction is "Jump to". If the instruction is "Jump to n", then begin doing instructions at number n. If there are no more instructions, stop.

Again we tested her knowledge by askeng her to type Program 3 and to predict the result:

Program 3:
```
010  Jump to 030
020  Print "Hi"
030  Print "Rochelle"
```

Student: The computer will go to the third instruction and print "Rochelle", then to the second and print "Hi", and then to the third again and print "Rochelle".

The actual result is:

```
Rochelle
```

Only one word is printed, and then the program halts. Why did the student predict what she did, when according to Schema 3 she should have been able to predict the result properly? Evidently she has other schemas about the operation of the computer. Many students seem to believe that every statement must be executed at least once, and this schema could apply here. If so, this causes a conflict with Schema 3, which might possibly be resolved by a reversion to one of the earlier schemas for JUMP-TO. Whatever the reason, it was a simple matter for the student to modify her schema for "Jump". When she saw that the output was the single word "Rochelle", she was readily able to

determine why:

Student: The first instruction tells it to go on to the third and then there is no instruction to tell it what to do so it stops.

Now, finally, she seems to have a complete and correct schema for the JUMP-TO instruction. When given two more tests, she predicted the results correctly.

Program 4: 010 Print "Hi"
020 Print "Rochelle"
030 Jump to 020

The predicted and correct result is:

HiRochelleRochelleRochelleRochelleRochelleRochelleRochelleRochelleRochelleRoch

This shows that the student does not believe that each repetition needs to be the same.

Program 5: 010 Print "Hi"
020 Jump to 010
030 Print "Rochelle"

The predicted and correct result is:

HiHiHiH

Our observations of the learning process, and especially our tutorial studies of the American Civil War and of the programming language FLOW indicate that students do not learn new structures by simply taking the new knowledge and adding it to what they have already acquired. Rather, the process appears to be a more active synthesis, a filtering of the new material through the data structures already existing, changing and modifying both new and old material to make some structure that the student finds satisfactory. The tutor's goal is to understand what the student is doing, how the student is interpreting the new information. The tutor provides sufficient background material that the new material can be properly interpreted. In addition, since new knowledge is of little use if it cannot be retrieved when needed, some attention must be paid to factors that influence memory retrieval. Several of our studies were concerned with these issues.

Experimental Studies of the Learning Process

In an effort to understand the processes involved in complex learning, we have studied subjects learning Civil War history and hypothetical maps. The following three sections summarize our findings from experiments in three separate but closely related projects.

Civil War Tutorials

In the examination of learning through individual tutorials, we have engaged in several long-term tutorial projects. We nave taken individual students and followed the evolving state of their knowledge as they read and discussed material about the Civil War. One basic text was used for their lessons: The Golden Book of the Civil War (1961). This text was stripped of all illustrations and unnecessary chapters were deleted. The result was a 64-page document of approximately 40,000 words. Subjects took from 5 to 11 hours to read the text, so that reading speed ranged between 60 to 120 words per minute. These slow rates reflect the complexity of the material and (for selected subjects) the time spent examining maps of the theater of war.

One major exploration concerned the use of subsidiary material by the learners. The text was entirely self-contained so that theoretically no extra material should have been needed. But all discussions of strategies and battles were strictly verbal since all maps and illustrations were removed. Without maps, as we predicted, good spatial representations of the events could not be formed.

Compare the performance of two subjects, one who was allowed free use of maps (Subject WM: With Maps), and one who was allowed only a small map showing the locations of all the states (Subject NM: No Maps). The differences are shown in Table 3.

Table 3. Recall of the Incidents of the Civil War after Study with and without Maps

	With Maps	No Maps
Overall Accuracy	0	0
Global Understanding (conceptualization)	+	-
Temporal Accuracy		
Locally	+	0
Globally	-	+
Detailed Knowledge	0	+

+ means superior recall

0 means satisfactory recall

- means inferior recall

NM tended to recite the battles in temporal order without regard for their locations. WM split the war into several different campaigns: the eastern, the western rivers, the southern port cities. WM tended to get local sequences more accurately than did NM, and to understand the motives behind the incidents. NM tended to intermix battles that had occurred at adjacent times, but over 1000 miles apart. NM had a higher accuracy of the ordering of events in general, but erred in presenting local sequences. NM could correct errors in WM's performance, but thought of the entire war as one single compaign. Thus, WM had a better conceptual grasp, but had poor knowledge of the time frame across areas of the country. NM had a better grasp of details, but a worse conceptualization.

When subjects learn complex material, they must integrate different sources of information. For the Civil War, these would include:

Identities of the major characters

The political situation

The economic situation and policies of the two sides

The industrial situation

Transportation routes (waterways and railroads)

Geography (and its effect on transportation routes and weather)

Relative size and power of armies and navies

None of these dimensions of information seems critical to the understanding of the war, but each adds its share. Each source of information provides some help in integrating the various events, and adds an important source of organization of the information. From our experiences with these tutorials, we conclude that multiple sources of information are important. A topic as complex as war history requires the integration of many different aspects. Even when the relationships are not direct, subsidiary information seems useful in allowing the whole to form a tighter organizational structure.

Studies on the Organization and Recall of Text

In a separate series of experiments on how people learn, organize, and store complex bodies of information, we examined the recall of narrative prose for evidence of underlying representational structures such as those proposed by Rumelhart's story grammar. (Reported in Gentner, 1976; in press.) According to the grammar, stories are decomposed into smaller units, such as settings, episodes, actions, reactions, events and goals. These smaller units (which may be further decomposed) are interrelated with causal predicates.

Subjects listened to tape recordings of two pages from a history text describing the first phase of battles along the northern Mississippi area

(to the battle of Shiloh). Their developing knowledge state was followed by collecting recalls after each presentation of the prose passage.

For analysis, the passage as divided into 143 "facts" which were organized according to two different structures: 1) a serial structure based on the serial order of the facts in the passage; 2) a story grammar structure based primarily on the causal relations among the facts. The analyses of the recalls showed that both of the structures are important for the memory of the passage. We looked at how the number of "neighbors" of a fact in a given structure which were mentioned on Trial n affect the recall of that same fact on Trial n+1. (The neighbors of a fact are the facts adjacent to it in the structure. In a serial organization, each fact has two neighbors. In a story-grammar organization, each fact may have any number of neighbors greater than one.)

In general, as a fact has more neighbors mentioned, it is more likely that the fact will be remembered correctly, and less likely that it will be forgotten, on the next recall. However, there are important differences in the effects of the serial and story-grammar structures. When a fact is absent on Trial n, the number of neighbors in both the serial and story-grammar structures mentioned on Trial N affects the recall of that fact on Trial n+1, but the number of neighbors in the serial structure has the greater effect. When a fact is partially-correct on Trial n, again neighbors in both structures mentioned on Trial n affect the recall of that fact on Trial n+1, but now the number of story-grammar neighbors mentioned has the greater effect. Finally, if a fact is correct on Recall n, its recall on Trial n+1 is influenced only by the number of its neighbors in the story-grammar

structure also mentioned on Trial n; the number of its neighbors in the serial structure mentioned on Trial n does not have any effect.

This pattern of results has a simple explanation. On first hearing the tape recording, the subjects perceive the passage as a collection of sentences or facts strung together in serial order, but as postions of the passage begin to "make sense," they perceive and organize the passage in a manner closer to its underlying meaning structure, and the serial order loses its initial importance. We found that only the neighbors in the story-grammar structure actually mentioned by the subject on one recall will affect performance on the subsequent recall. The structure inherent in a prose passage has no effect unless it is present in the subject's memory for that passage.

Studies on Map Learning

Our studies of the learning of Civil War history indicated that spatial information played an important role in the conceptualization of the campaigns and the individual battles. Accordingly we became interested in how people learn spatial information, particularly maps, and how they integrate spatial information with verbal information. The experimental studies concentrated on two areas: 1) The process by which maps are learned: Is there a general strategy which people use when they study a map? 2) The processes of storage and retrieval of map information: Are there particular organizations which are more effective than others for the storage and retrieval of map information? (This work is reported in detail in Shimron, 1975.)

The map used in these experiments contains a river, a mountain range, three major roads, two bridges, and ten cities (Figure 10). A city was located at each road intersection and bridge and the major roads were named. Thus, the map contained both spatial and verbal information.

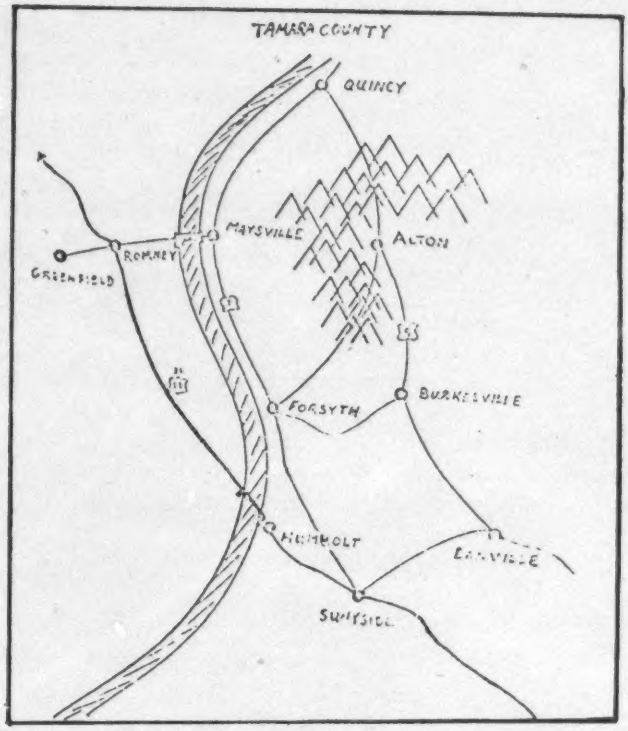


Figure 10. Experimental map (reduced 60%).

In the first experiment, one group of subjects was given unlimited time (approximately 12 minutes) to study the map, while another group of subjects was allowed to study the map for a limited time (6 minutes). On a subsequent recall test the two groups gave a similar performance on questions involving local organization (e.g., Which city is in the mountains?), but the subjects with unlimited time did significantly better on questions involving global organization (e.g., Name the cities on Route 7 going south).

In the second experiment, the subjects received the map information in three consecutive, cumulative presentations. One group received one dimension of information (e.g., the mountains and the river) in the first presentation, an additional kind of information in the second (e.g., the highways), and the rest of the information (e.g., city locations) in the third presentation. The second group received one section of the map (e.g., the northern third) in the first presentation, an additional section in the second presentation, and the remaining section of the map in the third presentation.

There was no difference in the performance of the two groups on most questions in the recall test, but the subjects who received the map in sections did significantly better on some of the questions involving global organization. The results indicate that while identifying different dimensions of information may be an important strategy in learning maps, optimal associations between these dimensions are not formed unless they are presented simultaneously.

The third experiment investigated other activities which could facilitate the learning of maps. Two groups of subjects were given a map to study for about 3 minutes. In addition, one group was asked to copy the map on a blank sheet of paper, while the second group was asked to read two stories which had the map as the background of the events described. The time for the two tasks was equal (about 6 minutes). On the questions in the subsequent recall test where there were significant differences, the subjects who read the stories did better than those who copied the map.

Map information seems to be organized in several ways: 1) by the general map concepts or schemes, such as highways, rivers and mountains; b) by identifying familiar shapes and associating locations with it; c) by locating shapes and schemes within the coordinates of the map; d) by encoding relations which specify how the schematic organizations are correlated.

The results of the experiments suggest the following principles of learning: 1) Learning a map is a gradual process whereby local connections between map elements are learned first and overall integration of the map

units is only achieved later; 2) The learning of map information is facilitated when the learner can observe how map units are simultaneously organized: By contrast, those who learn the map scheme-by-scheme (e.g., the highways separately from the river and the mountains) may have difficulties in relating one kind of schematic organization to another; 3) Propositional networks, such as stories which are relevant to the map, help to organize and to retain map information in memory.

Summary

Research under this contract period has centered around studies of the representation of knowledge in memory and studies of the learning and teaching processes. The active structural network, a type of semantic network, has been used to represent both knowledge and procedures within the same data base. Information in the active structural network is represented as propositions in the form of concepts interconnected by directed relations. This approach works well for local knowledge, but does not capture some of the higher level structure which can be found in narratives such as American Civil War history. The event schema representation has been developed to overcome this problem. It deals directly with the way that people organize complex events. Event schemas have been used to structure Civil War tutorials and to study the acquisition of concepts and error recovery as students learn computer programming. Schemas also are being applied to guide the automated FLOW tutor to store knowledge, to solve problems and to interpret student behavior.

The studies of the learning and teaching processes have been primarily concerned with developing models of the student and with the building of automated tutorial systems. The automated tutorial systems have provided a controlled environment in which to observe learning and teaching. Two contrasting topic matters have been used in the tutorial systems, Civil War history and computer programming. The Civil War tutorials convey a large body of factual information, while the FLOW programming tutorials are concerned with procedural and problem solving skills.

The experimental procedure of "incremental simulation" in which the computer performs whatever aspects of the tutorial project can be automated

and human tutors play the role of other components of the system, has proven to be an effective method of proceeding. This technique permits an emphasis on the understanding of the interactions among components of the overall system. This is important because it allows ideas to be tested without having to be concerned about all the details of the system, and without exerting the time and effort required for full simulation. Changes in procedure are possible sooner than would be possible were full simulation always required.

Our studies indicate some simple rules a teacher might follow to present material to a student. Basically, one wants to provide the learner with some sort of structure that will help organize what is to be acquired. In addition, one wants to provide multiple organizations and multiple viewpoints. This allows for redundant coding, causing an increased likelihood of later memory retrieval. It also allows the information that is sought to be derived from the constraints imposed by all that has been recalled up to a certain point. Even if these structures are partially erroneous and highly oversimplified, they can be useful learning tools.

References

Bobrow, D.G., & Norman, D. A. Some principles of memory schemata. In D.G. Bobrow & A.M. Collins (Eds.), Representation and understanding: Studies in cognitive science. New York: Academic Press, 1975.

Brown, J.S., Burton, R.R., & Zdybel, F. A model driven question-answering system for a CAI environment. Technical Report No. 13, January 1972. Department of Information and Computer Science, University of California, Irvine.

Carbonell, J.R. AI in CAI: An artificial-intelligence approach to computer-assisted instruction. IEEE Transactions on Man-Machine Systems, December 1970, Vol. MMS-11, No. 4, 190-202.

Gentner, D.R. The structure and recall of narrative prose. Technical Report, October 1975. Center for Human Information Processing, University of California, San Diego. (Also to appear in the Journal of Verbal Learning and Verbal Behavior, 1976, in press.)

Gentner, D.R., Wallen, M.R., & Miller, P.L. A computer-based system for studies in learning. Technical Report, September 1974. Center for Human Information Processing, University of California, San Diego.

The Golden book of the Civil War. New York: Golden Press, 1961.

MacDonald-Ross, M. The problem of representing knowledge. Paper presented at the Third Annual Conference on Structural Learning, Philadelphia, 1972.

Mehan, H. Language using abilities. Language Sciences, October 1972.

Norman, D.A. Cognitive organization and learning. Technical Report, August 1973. Center for Human Information Processing, University of California, San Diego.

Norman, D.A., Gentner, D.R., & Stevens, A.L. Learning and teaching. Technical Report, March 1974. Center for Human Information Processing, University of California, San Diego.

Norman, D.A., Gentner, D.R., & Stevens, A.L. Schemata and memory representation. In D. Klahr (Ed.), Cognition & Instruction. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1976.

Norman, D.A., Rumelhart, D.E., and the LNR Research Group. Explorations in cognition. San Francisco: W.H. Freeman, 1975.

Pask, G. The caste system. 1971. Systems Research, Ltd. Reported in MacDonald-Ross (1972).

Pask, G., & Scott, B. Uncertainty regulation in learning applied to procedures for teaching concepts of probability. 1972. Systems Research, Ltd. Reported in MacDonald-Ross (1972).

Quillian, M.R. Semantic memory. In M. Minsky (Ed.), Semantic information processing. Cambridge: MIT Press, 1968.

Rumelhart, D.E. Notes on a schema for stories. In D.G. Bobrow & A.M. Collins (Eds.), Representation and understanding: Studies in cognitive science. New York: Academic Press, 1975.

Shimron, J. On learning maps. Technical Report, October 1975. Center for Human Information Processing, University of California, San Diego.